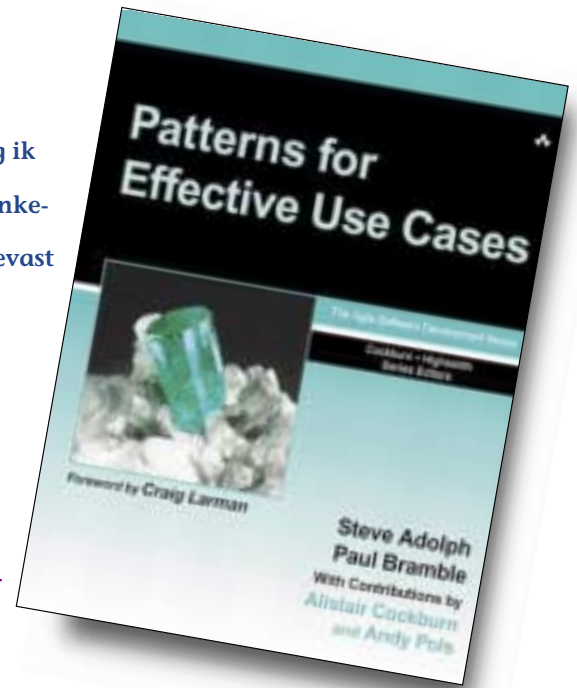


Tijdens lezingen en seminars waar ik het onderwerp adresseer vraag ik graag wie er gebruik maakt van de modelleertechnieken van UML. Onafhankelijk van waar ik de vraag stel, in binnen- of buitenland, is het antwoord stevast hetzelfde. Use cases zo'n veertig procent. Klassediagrammen ongeveer een derde. Daarna wordt het snel minder. Sequence diagrammen en activity diagrammen zo nu en dan. En dan is het stil.

PuzzleOutPattern Book



Erg? Helemaal niet. UML kent diverse modelleertechnieken die de moeite van het bestuderen nauwelijks waard zijn, uitgezonderd in zeer gespecialiseerde domeinen. En over de modelleertechnieken die wel gebruikt worden, kunnen we het maar moeilijk eens worden. Neem use cases als leidend voorbeeld. Zo doen er verwoede discussies de ronde over de granulariteit van use cases. Soms tref ik in projecten die al langer dan een jaar duren minder dan tien use cases. Zeg maar deelsystemen. Meestal varieert het aantal tussen de vijftig en de honderd use cases. De term wordt overigens met enige regelmaat gebruikt om een traditioneel functioneel ontwerp modern te laten klinken. Onder het mom dat projecten UML doen schrijft men dezelfde dikke pakken papier als pakweg tien jaar geleden. Alles is een use case. Toch?

Maar er zijn meer discussies over use cases. Is een use case nu een grafische techniek, zoals in UML, of vooral een tekstuele techniek? Persoonlijk vind ik de grafische modelleertechniek uit UML, aangevuld met kleine stukjes tekst erg praktisch. Er zijn ook methodologen die zweren bij alleen de tekstuele

representatie. Niet erg, als het doel maar hetzelfde is. Dat vindt ook Alistair Cockburn met wie ik onlangs over use cases sprak. Cockburn geldt als een van de autoriteiten op dit gebied. Hijzelf maakt nauwelijks tot geen gebruik van de grafische representatie van UML, maar schrijft vooral tekst, getuige zijn boek *Writing Effective Use Cases*.

Het boek *Patterns for Effective Use Cases* van Steve Adolph en Paul Bramble borduurt voort op dit eerdere werk. Het kent zelfs bijdragen van Cockburn. Het verschijnt in de Agile Software Development Series waarvan alweer Alistair Cockburn editor is. Ook Adolph en Bramble staan vooral stil bij de tekstuele representatie van de techniek. Het boek bevat dertig patronen die allemaal gerelateerd zijn aan de werkwijze waarop use cases tot stand komen, hun onderlinge samenhang en de wijze van beschrijven van use cases. Voor wie wel eens worstelt met de techniek is dit boek niet onaardig.

Het boek is zonder twijfel amusant en kent goede tips voor gevorderde lezers. *“Restrict the number of people refining any one work product to just two or three people.”* Een tip die ook veel discussies oplost is de vol-

gende. *“Write each use case to address one complete and well-defined goal. That goal may be at any level.”* Begrepen?

Maar ondanks deze goede tips bekruipt mij toch het gevoel dat het wel wat minder kan. Beide auteurs gaan wel erg uitgebreid in op de geboden patronen. Dat had korter gemogen. Bovendien leidt het boek aan het **PatternBookSyndrome**. Het beschrijft patronen aan de hand van namen van andere patronen. En dat leest bijzonder onprettig. Een voorbeeld? Sla patroon **CompleteSingleGoal** op en lees de eerste zin. *“You are working on the EverUnfoldingStory, focusing on a ClearCastOfCharacters.”* Een patronenboek is snel onleesbaar. Mijn advies? **DrinkCoffeeFirst** en **ReadTips** voor dat u **PuzzleOutPatternBook**.

Sander Hoogendoorn, Partner Ordina
(www.sanderhoogendoorn.com)

Titel: Patterns for Effective Use Cases
Auteur: Steve Adolph, Paul Bramble e.a.
Uitg.: Addison Wesley 2003